VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SADA TESTŮ PRO PROJEKT OPENLDAP/NSS

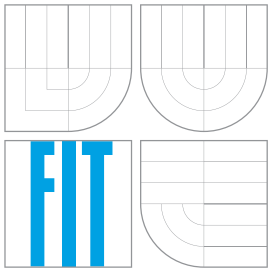
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DAVID ŠPŮREK

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SADA TESTŮ PRO PROJEKT OPENLDAP/NSS

A TEST SUITE FOR THE OPENLDAP/NSS PROJECT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DAVID ŠPŮREK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ALEŠ SMRČKA, Ph.D.

BRNO 2011

Abstrakt

Cílem práce je vytvořit sadu testů pro projekt OpenLDAP/NSS. Práce objasňuje principy testování, možnosti automatizace testování v prostředí GNU/Linux a vysvětluje pojmy LDAP i NSS. Testy jsou navrženy pro distribuci Fedora/Red Hat Enterprise Linux a jsou automatizovány pomocí skriptování v shellu s využitím knihovny Beakerlib pro testování. Pro návrh testů je použita metoda black box. Testy se zaměřují na integraci OpenLDAP a NSS. V práci jsou navrhnuty a implementovány testy pro balíky openldap-clients, kerberos, nss-pam-ldapd, samba a autofs.

Abstract

The goal of this thesis is to create a test suite for the OpenLDAP/NSS project. The work clarifies the principles of testing, automation testing in a GNU/Linux environment, and the concepts of LDAP and NSS. Tests are designed for Fedora/Red Hat Enterprise Linux distribution and they are automated using shell scripting and the Beakerlib library for testing. Black-box testing method is used to design tests. The tests focus on the integration of OpenLDAP and NSS. In the thesis, tests are designed and implemented for openldap-clients, kerberos, nss-pam-ldapd, samba, and autofs packages.

Klíčová slova

Fedora, LDAP ,OpenLDAP, NSS, testování software, black-box testování, Beakerlib, AutoQA

Keywords

Fedora, LDAP, OpenLDAP, NSS, software testing, black-box testing, Beakerlib, AutoQA

Citace

David Špůrek: Sada testů pro projekt OpenLDAP/NSS, bakalářská práce, Brno, FIT VUT v Brně, 2011

Sada testů pro projekt OpenLDAP/NSS

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Aleše Smrčky, Ph.D.

.....
David Špůrek
11. května 2011

Poděkování

Děkuji svému vedoucímu Ing. Aleši Smrčkovi, Ph.D. za rady a náměty při psaní práce. Chtěl bych poděkovat i firmě Red Hat Czech, s.r.o za možnost spolupráce a pomoc při realizaci testů. Konkrétně děkuji Ondřeji Hudlickému a Ondřeji Morišovi.

Obsah

Kapitola 1

Úvod

V dnešním světě plném počítačů a chytrých zařízení se můžeme setkat s velkým množstvím softwarových aplikací. Aplikace mohou pocházet od různých autorů a mít odlišnou úroveň kvality zpracování. Autorem programů je člověk, jenž není neomylný a může do software zanést chyby jak při tvorbě aplikace, tak při následné opravě již existujících chyb. Dnes se ve firmách pracuje v týmu, kde špatná komunikace mezi členy může způsobit chybu v programu. Při vývoji se může objevit problém u integrace několika aplikací do jednoho spolupracujícího programu. Tvůrce by měl počítat s variantou, že se může v software objevit chyba. Každá aplikace musí být řádně otestována a právě tato práce se bude zabývat testováním a hledáním chyb v aplikacích.

Integrační testování je jedním z typů softwarového testování, které se vyskytuje v testech navržených v mé práci. Cílem práce bude testování integrace OpenLDAP a externí kryptografické knihovny NSS v distribuci Fedora. Kryptografická podpora NSS se v distribuci Fedora používá od verze F14, dříve byla součástí kryptografická knihovna OpenSSL. Změna kryptografické knihovny by neměla mít vliv ani na klienta, ani na server (to, co předtím fungovalo s OpenSSL, by mělo fungovat i nyní bez nutnosti změny konfigurace). Testy jsou zaměřeny na ověřování šifrované komunikace mezi klientem a serverem. Šifrovanou komunikací se rozumí TLS nebo SSL. Testována je funkčnost a podpora šifrování, nikoliv síla zabezpečení.

Serverem je v testech OpenLDAP server s kryptografickou knihovnou NSS. Klienty představují různé balíčky dostupné v distribuci Fedora, které mohou používat šifrované spojení s LDAP serverem, jež jsou potřebná pro načtení dat a následnou činnost balíčku. Balíčky openldap-clients, kerberos, autofs, samba a nss-pam-ldapd jsou klienty v jednotlivých testech.

První část textu se zabývá principy testování a možnostmi automatizace testování v prostředí Linux. Následuje kapitola vysvětlující pojem LDAP, NSS a implementaci LDAP nazvanou OpenLDAP. Kapitola objasňuje i účel balíčků a jejich možné propojení s OpenLDAP serverem. Čtvrtá kapitola Plán testování se zabývá hlavní částí práce, která popisuje jednotlivé fáze testů pro vybrané balíčky. Navazující pátá kapitola popisuje, jak navržené testy automatizovat pomocí BeakerLib a jak je začlenit do AutoQA. Automatizované testy budou provedeny na ditribucích Fedora a Red Hat Enterprise Linux. Na konci této kapitoly je zhodnoceno pokrytí testů a navrhnuto možné vylepšení v dané oblasti. Na závěr práce budou zhodnoceny dosažené výsledky.

Kapitola 2

Softwarové testování

V této kapitole budou popsány přístupy k testování software a možnosti automatizace v prostředí GNU/Linux. Podkapitola přístupy k testování software objaňuje pojmy testovací metody a testovací úrovně.

2.1 Přístupy k testování software

Softwarové testování zjišťuje kvalitu, správnost a bezpečnost vyvíjeného software. Testováním lze získat informace o splňení požadavků zadaných při specifikaci. Softwarovým testováním není možné zaručit správnost a bezchybnost programu. Lze konstatovat, že za daných podmínek se aplikace chová správně nebo ne. Program nelze testovat pro všechny vstupní podmínky, kterých může být potenciálně nekonečno. Ze stejného důvodu není možné zkontrolovat správnost všech výstupních hodnot. Pro vývoj programu se zpravidla používá jeden z modelů vývojového procesu. Testování se vyskytuje v jednotlivých modelech v různých fázích. Největší úsilí na testování software je vynakládáno po dokončení implementační fáze.

2.1.1 Statické a dynamické testování

Techniky revize, procházení a inspekce kódu jsou zahrnuty do statického testování. Vykonávání programového kódu na množině daných testovacích případů je označováno jako dynamické testování. Dynamické testování může být zahájeno ještě před definitivním dokončením programu a mohou se testovat již dokončené části kódu [5].

2.1.2 Testovací metody

Nejčastěji používanými metodami jsou white box a black box (původně anglické termíny bez českého ustáleného výrazu). Obě metody patří do skupiny dynamického testování.

White box testování

Návrhář testů zná vnitřní strukturu programu. Ze znalosti programového kódu určí vstupní hodnoty a podmínky tak, aby test prošel programem vhodným způsobem a získal požadované výstupní hodnoty. Způsob White box testování může být uplatněn na jednotkové (anglicky unit), integrační (anglicky integration) nebo systémové úrovni procesu testování. Na jednotkové úrovni se uplatňuje nejčastěji.

Black box testování

Black box testování chápe testovanou aplikaci jako černou skříňku, u které nezná její vnitřní strukturu. U tohoto přístupu víme jen to, jaký výsledek dostaneme po použití různých vstupů. Tento způsob se používá při testování funkcionality programů. Jednotlivé testovací případy jsou vytvořeny na základě specifikace a požadavků, které by měla aplikace splňovat [14].

Přístup black box je použit k tvorbě testovacích případů vytvořených v rámci této práce.

2.1.3 Testovací úrovně

Software lze testovat jednou z mhoha metod a podobně je možné pro test zvolit jednu z řady testovacích úrovní. Každá úroveň se zabyvá testovaním s odlišným pohledem na implementaci. Pokud se na úrovně podíváme z pohledu procedurálního programování, může být úroveň zaměřena na funkci (Jednotkové testování), propojení a správné předávání hodnot mezi funkcemi (Integrační testování) nebo testování celého programu (Systémové testování).

Podobně jako softwarové testování nemůže pokrýt všechny varianty průchodu programem, ani jediná testovací úroveň nepokryje veškeré varianty vykonání programu.

Jednotkové testování

Jednotkové testování používáme pro ověřování nejmenších testovatelných úseků kódu, tzv. jednotek. Jednotkou může být, z pohledu procedurálního programování, funkce nebo procedura. Testy slouží k ověření správné funkčnosti jednotlivých elementárních částí kódu a v ideálním případě jsou na sobě nezávislé.

Testování částí kódu programátorovi umožňuje, aby se po čase mohl k jednotlivým testům vrátit a ujistit se, že daná část stále pracuje správně. Jestliže modifikace funkce nebo metody způsobí chybu, může být rychle identifikována a opravena [14].

Integrační testování

Ověřuje správnou funkčnost více modulů, seskupených a testovaných jako jeden celek. Tento způsob testování odhaluje problémy při interakci mezi rozhraními jednotlivých integrovaných modulů. Integrační testování většinou provádíme přístupem Black box.

Integrační úroveň testování je použita pro návrh testů v této práci, jelikož OpenLDAP server a externí kryptografickou knihovnu NSS lze vnímat jako dva moduly. Moduly jsou seskupeny v jeden celek a uživatel má možnost přistoupit k datům serveru pomocí šifrovaného spojení.

2.2 Možnosti automatizace testování v prostředí GNU/Linux

Pro automatizaci testování v prostředí GNU/Linux existuje mnoho nástrojů a 3 z nich budou popsány v této podkapitole.

Software Testing Automation Framework (STAF)

Následující popis byl převzat z domovské stránky projektu STAF [2].
Software Testing Automation Framework je projekt s otevřeným kódem a může běžet

na více počítačových platformách. Jedná se o framework (softwarová struktura sloužící k usnadnění programování jiných softwarových projektů, v češtině nemá tento termín ustálený výraz) navržený na základě myšlenky znovupoužitelnosti částí kódu a volání služeb (vyvolání procesu, řízení zdrojů, logování a dohlížení). STAF smazává problém s vytvořením struktury automatizace a tím umožňuje zaměření se na budování řešení vlastní automatizace.

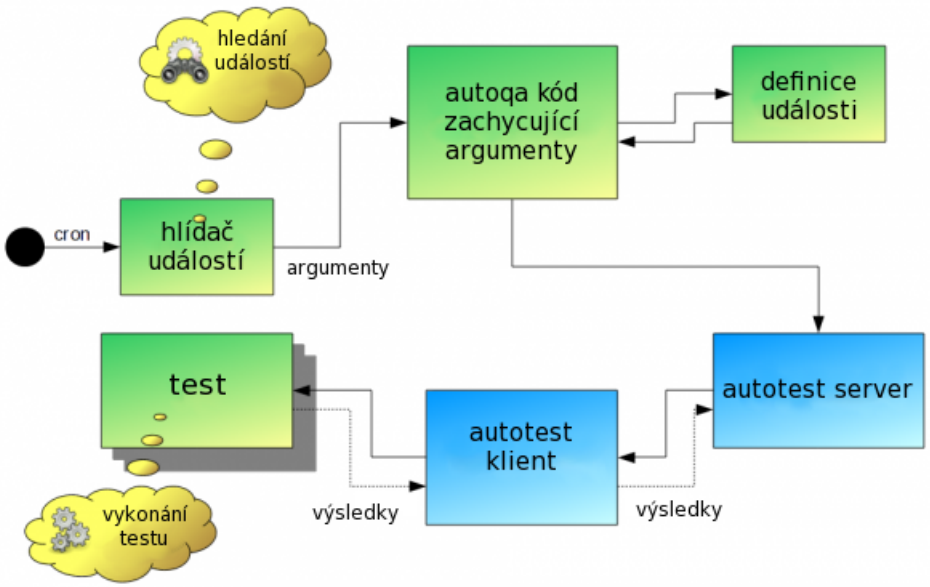
STAF pomáhá vyřešit problémy v průmyslu, jako je častější frekvence výrobních cyklů, kratší doba přípravy, snížení času na testování, možnost volby více platforem, více programovacích jazyků a zvýšených požadavků na národní jazyk. STAF může v těchto oblastech pomoci, protože je prokázáno, že použití vyspělé technologie napomáhá automatizaci.

Phoronix Test Suite

Phoronix Test Suite je testovací, srovnávací, platformně volitelný a rozšiřitelný framework, do kterého lze snadno přidat nové testy. Software je navržen tak, aby efektivně prováděl kvalitativní a kvantitativní srovnávací test čistým, opakovatelným a uživatelsky přívětivým způsobem [8].

AutoQA

Téma bakalářské práce bylo zadáno ve spolupráci s firmou Red Hat. Firma požadovala použití nástroje AutoQA, který bude popsán podrobněji v podkapitole 5.1. Automatizované testování poskytuje možnost samostatně spouštět testy a shromažďovat jejich výsledky. Návrh je jednoduchý - pokud se objeví definovaná událost, AutoQA spustí automatizované testy [10]. Princip AutoQA vidíme na obrázku 2.1.



Obrázek 2.1: Náhled na AutoQA architekturu, obrázek převzat z [12]

AutoQA definuje hlídače událostí, které jsou pro autora testu zajímavé. Hlídač se primárně skládá z těchto bodů [11]:

- definice uživatelsky zajímavých událostí (např. vydání nové verze balíčku)
- hlídač sleduje události a zjišťuje argumenty potřebné pro testy
- autoqa kód zachycující argumenty, které přicházejí od sledovacího programu
- seznam testů, jež se mají provést po zachycení události
- šablona pro napsání nového testu tohoto hlídače

AutoQA spolupracuje s Autotest framework. Autotest získává od AutoQA seznam testů, jež mají být spuštěny. Server autotest má k dispozici seznam klientů vykonávajících testy. Požadavky na spuštění testů řadí do fronty, ze které odebírají jednotlivé záznamy autotest klienti. Klienti následně testy vykonají a výsledek předají serveru.

AutoQA případy užití

Při definici případů užití věnujeme pozornost několika bodům, které definují, co bude testováno. Návrhář testů potřebuje mít seznam dostupných testů, jež mohou být opakovaně vykonány. Každý test má řídicí soubor (control file) obsahující i popis daného testu. Dalším bodem je definice seznamu hlídačů, určující jaké události monitoruje. Každý hlídač má přiložen soubor readme s jeho popisem a soubor se seznamem testů, vykonávaných při výskytu události [11].

Kapitola 3

OpenLDAP/NSS

Kapitola se zabývá vysvětlením principů LDAP a NSS. Po vysvětlení principů je popsána jejich implentace v projektu OpenLDAP/NSS. V další části kapitoly je objasněno, k jakému učelu slouží balíčky vybrané pro testování. Vybranými balíčky jsou openldap-clients, krb5, nss-pam-ldapd, samba a autofs. Na závěr kapitoly je navrženo propojení těchto balíčků s OpenLDAP/NSS a současný stav jejich testování.

3.1 LDAP (Lightweight Directory Access Protocol)

LDAP je adresářovou službou používanou pro zisk dat, majících charakter adresáře. Adresář lze vnímat jako databázi adres, telefonních čísel nebo emailových adres. LDAP v porovnání s databází neočekává častou změnu dat, nejčastější operací je vyhledání. Mírné neaktuálnosti u adresáře nevadí, u databáze by šlo o velký problém. Adresář je přizpůsoben pro jednoduchý přístup z mnoha aplikací (např. klient elektronické pošty), krátký čas odezvy a přehlednost. Administrátor spravující adresář požaduje spolupráci s dalšími aplikacemi, možnost distribuce dat na více serverů, decentralizované a hierarchické uspořádání informací (použit model DNS pro jmenný model), jednoduchou správu dat a kontrolu přístupu.

V této podkapitole bude popsán vznik a koncepce LDAP. LDAP popisují čtyři modely a každému z nich bude věnována samostatná část.

3.1.1 Vznik LDAP

Koncept LDAP vychází ze standardu adresář X.500. Jedná se o standard protokolů a informační model pro globální adresářovou službu nezávislou na výpočetním prostředí a architektuře. Podobně jako u DNS zahrnuje virtuální adresář celý svět.

Standard definuje reprezentaci dat v záznamech, organizaci dat a jejich jména – hierarchický model jmen DIT, model přístupu, správy a údržby dat. Standard znevýhodňuje větší množství protokolů, znesnadňujících implementaci i běh klienta a serveru. Definuje čtyři protokoly: přístup k datům (DAP), předávání dotazů mezi servery (DSP), replikace vybraných informací (DISP) a automatické ustavení spojení mezi servery (DOP).

3.1.2 Koncepce LDAP

LDAP byl navržen jako alternativa k X.500. Došlo ke zjednodušení implementace a použití jediného přenosového protokolu – LDAP. Stejně jako u X.500 se používá koncept DIT (Directory Information Tree - stromová struktura) pro uspořádání dat.

Čtyři modely popisují architekturu LDAP:

- Informační model – popisuje uložení informací v adresáři
- Jmenný model – hierarchické uspořádání dat
- Funkční model – přístup k datům, operace protokolu LDAP
- Bezpečnostní model – zabezpečení informací v adresáři

3.1.3 Informační model

Základní jednotkou pro uložení informace je záznam popsaný pomocí třídy objektů a obsahující seznam atributů ve tvaru typ, hodnota. Každý záznam obsahuje jednoznačný identifikátor DN (Distinguished Name), definující jeho umístění ve stromu DIT.

Atribut definuje záznam tvaru typ, hodnota. Uvádí i způsob porovnání hodnot a zaručuje splňení syntaxe (danou typem) u hodnot.

Příklad záznamu ve formátu ldif:

```
dn: dc=my-domain,dc=com
objectClass: dcObject
objectClass: organization
dc: my-domain
o: my-domain
description: my-domain

dn: cn=Manager,dc=my-domain,dc=com
objectClass: organizationalRole
cn: Manager
description: Directory Manager
```

3.1.4 Jmenný model

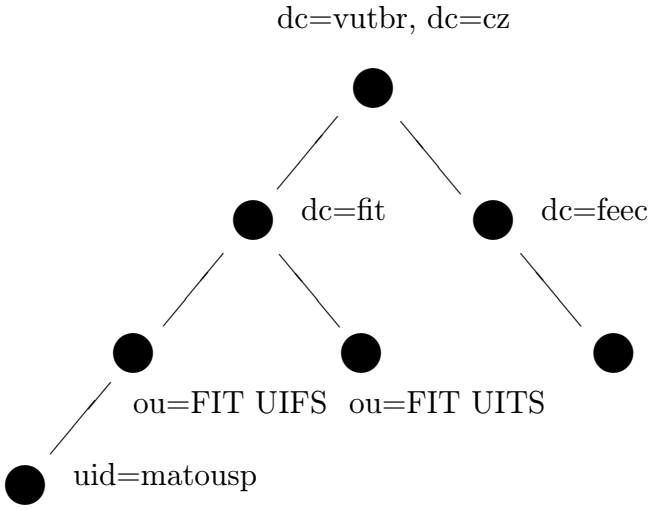
Tvoří stromovou strukturu DIT (obrázek 3.1) a definuje organizaci a identifikaci záznamů v adresáři. Popisuje strukturu adresáře vybudovanou z jednoznačných záznamů pojmenovaných pomocí DN, které strukturou připomínají DNS adresu.

Vrcholy stromu představují záznamy v adresáři a hrany symbolizují vztahy mezi záznamy.
Záznamy typu alias mohou odkazovat na jiný DN (podobné symbolickým linkům). Vyhledání takové záznamu bude náročné a je potřeba to vzít v úvahu.

3.1.5 Funkční model

Funkční model popisuje komunikaci postavenou na výměně zpráv. Komunikace je typu klient – server. Klient typicky vytvoří dotaz a zašle jej LDAP serveru. Server odpoví jednou nebo více zprávami obsahujícími odpověď. Implementace je provedena nad TCP vrstvou. Protokol LDAP definuje komunikaci, ale ne formát uložení dat.

Nejčastější operací adresáře je vyhledávání. U vyhledávání se specifikuje: začátek vyhledávání, rozsah a maska. Začátek je zadán ve formě DN, rozsah je buď záznam, jedna úroveň nebo podstrom. Maska typicky označuje hledaný výskyt atributu.



Obrázek 3.1: Ukázka stromové struktury, obrázek převzat z [13]

Dalším krokem je operace vytvoření spojení se serverem (bind), u kterého je nutné přihlášení, dohodnutí autentizace a identifikace uživatele. Po dokončení požadovaných operací se provede ukončení spojení (unbind). Adresář poskytuje operace pro práci se záznamy. Operacemi mohou být přidání, smazání, modifikaci, porovnání záznamu, změnu jednoznačného jména nebo zrušení předchozí operace.

3.1.6 Bezpečnostní model

Bezpečnostní model poskytuje ochranu dat adresáře před neoprávněným přístupem. Definuje kontrolu přístupu k adresáři. Podle zabezpečení se LDAP servery dělí na veřejné (anonymní přístup k datům pouze pro čtení), na servery s možností autentizace pomocí hesla (implementace MD5 SASL) a na nejbezpečnější servery s autentizací a kryptováním (operace pro TLS a SSL autentizaci pomocí veřejných klíčů a certifikátů) [13].

3.2 NSS (Network Security Services)

NSS představuje množinu knihoven navržených pro podporu meziplatformního vývoje zabezpečených klientských a serverových aplikací. Knihovny poskytují kompletní implentaci projektu s otevřeným kódem obsahující kryptovací standardy SSL v2 a v3, TLS, PKCS #5,#7,#11 a #12, S/MIME a X.509 v3 certifikáty.

Pokud aplikace potřebuje podporu bezpečnostních standardů SSL, S/MIME a dalších, může využít NSS. NSS je dostupné pod licencemi Mozilla Public License a GNU General Public License [6].

3.2.1 SSL

Následující popis byl převzat z internetové stránky projektu MDN (Mozilla Developer Network), která vysvětluje SSL [9].

Přenos dat po internetu se nejčastěji řídí protokolem TCP/IP. Ostatní protokoly jako HTTP, LDAP, IMAP a jiné běží nad tímto transportním protokolem. SSL umožňuje serveru (podporujícího SSL), aby se autentizoval klientovi (musí rovněž podporovat SSL). Autentizací klienta na serveru může dojít k vytvoření šifrovaného spojení mezi klientem a serverem. SSL autentizace serveru poskytuje možnost potvrzení identity serveru. Klient s podporou SSL může použít standardní techniky šifrování pomocí veřejného klíče a ověřit certifikát serveru vydaný certifikační autoritou (CA), kterou má klient v seznamu důvěryhodných certifikačních autorit. Takové ověření může být důležité například pro uživatele, který posílá číslo své kreditní karty.

Na druhé straně SSL autentizace klienta dovoluje serveru ověřit identitu uživatele. Server podporující SSL může ověřit certifikát klienta vydaný CA, kterou má server v seznamu důvěryhodných CA. Ověření identity klienta může být důležité například v případě, kdy chce banka odeslat důvěrné finanční informace zákazníkovi a chce ověřit jeho identitu.

Šifrované SSL spojení vyžaduje aby všechny informace posílané mezi klientem a serverem byly zašifrované odesílajícím softwarem a dešifrované přijímajícím softwarem. Všechna data zaslaná přes šifrované SSL spojení jsou chráněny mechanismem detekce změny přenášených dat.

3.2.2 TLS

Šifrované spojení je možné vytvořit i pomocí TLS, které je založeno na SSL a funguje na podobném principu. Na rozdíl od SSL se nešifruje celé spojení, ale jen obsah probíhající komunikace.

3.3 Implementace OpenLDAP/NSS

OpenLDAP je volně šiřitelná implementace LDAP protokolu. Software se skládá ze tří hlavních částí – `slapd` (LDAP démon), spojený s pokrytím funkčnosti a nástroji pro LDAP server. Druhou částí jsou knihovny obsahující implementaci LDAP protokolu. Poslední částí je klientský software implementující funkce pro práci s adresářem.

OpenLDAP implementuje LDAP protokol verze 3 a podporuje Ipv4 i Ipv6. Slapd je možné nastavit podle konkrétních požadavků v jediném konfiguračním souboru. LDAP server umožňuje autentizaci a zabezpečení datových služeb pomocí SASL (Simple Authentication and Security Layer) nebo na základě certifikátů (zabezpečení použitím TLS nebo SSL). Šifrované spojení používá externí kryptografickou knihovnu. Distribuce Fedora od verze F14 používá kryptografickou podporu Mozilla NSS namísto OpenSSL. Komunikaci mezi OpenLDAP serverem a klienty realizuje knihovna libldap. Šifrovaná komunikace je z pohledu serveru a klientů černou skříňkou (je to věcí libldap knihovny). Pojmem šifrovaná komunikace se rozumí SSL nebo TLS. SSL šifruje celé spojení, před doménové jméno nebo IP adresu OpenLDAP serveru přídává `ldaps://` a připojuje se na port 631. TLS narozdíl od SSL šifruje pouze obsah komunikace a před doménové jméno nebo ip adresu OpenLDAP serveru přídává `ldap://`. Spojení je navázáno na standardní port 389 stejně jako v případě nešifrované komunikace.

3.4 Balíčky pro OpenLDAP/NSS dostupné v rámci distribuce Fedora

Podrobněji budou popsány jen některé balíčky z distribuce, jež mohou využívat služeb OpenLDAP serveru. Pro tyto vybrané balíčky jsou dále v této práci vytvořeny testy. Přehled balíčků, které mohou využívat LDAP, lze najít na [1].

3.4.1 Balíček openldap-clients

Balíček openldap-clients je nedílnou součástí openldap projektu. Poskytuje prostředky pro přístup a práci s openldap serverem. Balíček se skládá z 10 menších aplikací. Každá aplikace poskytuje uživateli prostředek pro přístup a manipulaci s ldap serverem určitým způsobem. Účel těchto programů plyne již z jejich názvů. Openldap-clients poskytuje aplikace `ldapadd` (přidání záznamů), `ldapcompare` (porovnání záznamu-odpověď typu ano/ne, zda záznam obsahuje zadaný atribut s hodnotou), `ldapdelete` (smazání záznamů), `ldapexop` (rozšiřující LDAP operace), `ldapmodify` (změna záznamu-přidání, editace, smazání atributů), `ldapmodrdn` (změna relativního DN LDAP-přemístění záznamu v adresáři nebo změna jména záznamu), `ldappasswd` (změna hesla uživatele), `ldapsearch` (vyhledávání v LDAP adresáři), `ldapurl` (formátování LDAP URL ze zadaných atributů sestaví LDAP URI nebo naopak LDAP URI rozdělí na konkrétní atributy), `ldapwhoami` (zjišťuje s jakým jménem navázal uživatel spojení se serverem).

3.4.2 Balíček krb5

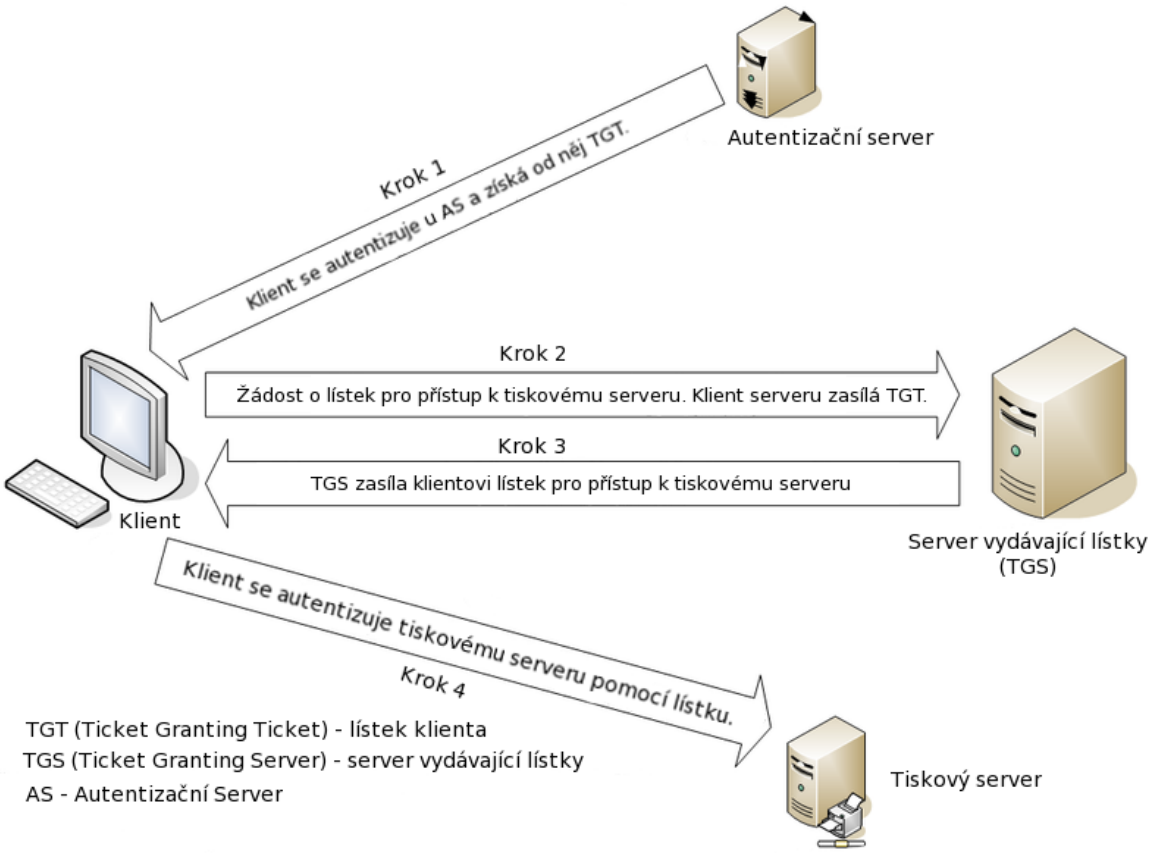
Kerberos je autentizační systém založený na zcela jiném principu, než běžné posílaní hesla po síti. Princip využívá posílání lístků namísto hesla, což je výhodné pro použití v nezabezpečené síťi. Autentizace se provádí pomocí zasílaní lístků požadované službě. Kerberos používá důvěryhodnou třetí stranu - KDC (Key Distribution Center), skládající se ze dvou částí. Částmi KDC jsou autentizační server (AS) a server vydávající lístky (TGS). Přístup k autentizačnímu serveru se obvykle odehrává pouze jednou v rámci relace. Princip autentizace pomocí Kerberos znázorňuje obrázek 3.2. Klient se autentizuje u AS a získá od něj TGT (Ticket Granting Ticket), pomocí kterého může žádat o lístky u TGS. Pokud chce klient přistoupit k nějaké službě, podporující autentizaci pomocí Kerberos, požádá pomocí TGT lístku TGS server o vydání lístku pro požadovanou službu. Pomocí tohoto lístku se klient následně službě autentizuje a může ji používat.

3.4.3 Balíček nss-pam-ldapd

Balíček je NSS (Name Service Switch) a PAM (Pluggable Authentication Module) modul, dovolující LDAP serveru poskytnutí uživatelských účtů, skupin, aliasů, síťových skupin, názvu hostitele a v podstatě jakékoli jiné informace běžně dostupné z normálních souborů v `/etc` nebo NIS. Umožňuje klientovi i autentizaci na LDAP serveru [3].

3.4.4 Balíček samba

Samba umožňuje sdílení dat mezi vzdálenými systémy. Vzdálené systémy dokáže připojit k lokálnímu systému a uživateli se systémy jeví jako lokální. Samba sdílí soubory, tiskárny a jiné prostředky mezi operačními systémy Windows a UNIX. Architektura je vystavěna na modelu klient/server. Možnosti služby Samba jsou založeny na dvou démonech. Smbd



Obrázek 3.2: Princip Kerberos, obrázek převzat z [4]

server komunikuje pomocí protokolu SMB. Umožňuje sdílení souborů a tiskáren, poskytuje i autentizaci a autorizaci. Nmbd provádí mapování NetBIOS na IP adresy pro klienty služby. Uživatelé UNIXu mohou přistoupit k Samba serveru pomocí nástroje `smbclient`, který se chová podobně jako FTP server.

3.4.5 Balíček autofs

Autofs slouží k automatickému připojování (mount) a odpojování (umount) výměnných médií a síťových svazků. Vytváří virtuální adresáře, do kterých podle potřeby připojuje nebo odpojuje svazky výměnných či síťových médií. Uživatel může zadat příkaz `ls` s cestou do takového virtuálního adresáře. V tu chvíli autofs podle konfigurace zjistí kde se nachází příslušný svazek a připojí jej v případě potřeby (běh příkazu `ls`). Pokud není svazek po nějakou dobu používán, dojde k jeho odpojení. Uživatel tak nemusí připojovat a odpojovat svazky, nepotřebuje znát ani podrobnosti o umístění svazku [7].

3.5 Případy užití a propojení těchto komponent s OpenLDAP/NSS

Balíček openldap-clients

Openldap-clients je základním nástrojem pro přístup k Openldap serveru. Propojení této komponenty se serverem je jednoznačné. Bude ověřeno spojení každé aplikace z tohoto balíčku se serverem. Vyjímkou je aplikace ldapurl, která nenavazuje spojení a není možné jako u ostatních aplikací testovat integraci kryptografické knihovny NSS. Pro úplnost testů bude do návrhu zařazena.

Balíček krb5

Kerberos KDC standardně ukládá záznamy ve své vlastní databází. Namísto vlastní databáze lze pro uložení záznamů použít LDAP server. Použití LDAP jako databáze pro záznamy lze nastavit v `/etc/krb5.conf` v sekci `[dbmodules]`.

Balíček nss-pam-ldapd

Umožňuje získávat informace o uživatelích a skupinách nejen z lokálních souborů, ale i z LDAP serveru. Musí být správně nastaveno `/etc/nsswitch.conf`, které definuje kde se má dotazovat na požadované informace. Jedním z případů užití může být dotaz na heslo uživatele, jehož záznam je uložen v LDAP. Takový dotaz může být zadán příkazem `getent passwd jméno_uživatele_v_ldap`.

Balíček samba

Uživatelé musí mít někde definovaná uživatelská jména a hesla pro přístup, aby se mohli uživatelé připojit k Samba serveru. Typicky bývají uloženy v lokálních souborech v `/etc`. Tyto informace je možné uložit i do LDAP adresáře.

Balíček autofs

Konfigurace autofs, zjišťující informace o svazcích a médiích, je typicky uložena v lokálních souborech `/etc/auto.master`, `/etc/auto.misc` (případně v dalších). Soubor `/etc/auto.master` určuje jaké virtuální adresáře se budou připojovat. Pro každý adresář poté existuje samostatný soubor s konfigurací. Pokud se má automaticky připojovat adresář `/misc`, pak existuje jeho konfigurační soubor `/etc/auto.misc`.

Pomocí nastavení v `/etc/nsswitch.conf` je opět možné získávat tyto informace i z LDAP serveru. Záznamy pro automatické připojování a odpojování jsou poté uloženy v LDAP serveru, místo uložení v lokálních souborech.

3.6 Současný stav testování balíčků OpenLDAP/NSS

Balíčky z projektu openldap jsou testovány pomocí upstreamu. Spolu se zdrojovými kódy openldap je distribuován i soubor vlastních selftests vytvořených vývojáři. Soubor obsahuje testy pro openldap-servers a openldap-clients. Podobně mají upstream testy i balíčky samba, nss-pam-ldap a kerberos. Důležité je zmínit, že upstream testy netestují SSL/TLS komunikaci. Vyjímkou jsou upstream selftests pro openldap.

Kapitola 4

Plán testování

Testy jsou navrženy pro balíčky, které mohou využít služeb OpenLDAP a jsou dostupné v distribucí Fedora nebo RHEL (Red Hat Enterprise Linux). Jedná se především o integrační testy OpenLDAP a kryptografické knihovny NSS nebo o testy ověřující správnou funkčnost šifrovaného přístupu k serveru. Důvodem pro vytvoření těchto testů je změna kryptografické knihovny od verze F14, kdy NSS knihovna nahradila zastaralou knihovnu OpenSSL. Tato změna kryptografické podpory by neměla mít vliv na funkčnost OpenLDAP/NSS a vše by mělo fungovat stejně jako dřív bez změny v nastavení.

Každý balíček bude napřed otestován *bez šifrování* pro ověření správné konfigurace LDAP serveru i balíčku. Následuje konfigurace balíčku *pro komunikaci přes TLS* a ověření správné funkčnosti, pokud daný balíček komunikaci přes TLS podporuje. Stejný postup je aplikován i *pro test komunikace přes SSL*.
Tyto 3 konfigurace balíčků jsou opětovně testovány s novým nastavením LDAP serveru, které vyžaduje ověření certifikátu klienta. Popis jednotlivých testů je rozdělen do tří částí: příprava testu, akce testu a očekávaný výsledek. Určité části přípravy nebo akce testu se opakovaně využívají v některých dalších testech.
Konfigurační soubory s konkrétními hodnotami jsou k nalezení v příloze.

4.1 Balíček openldap-clients

Testy jsou vytvořeny pro různé aplikace, různé nastavení šifrování komunikace a různé způsoby stejného nastavení. Každý test má svůj název, který je tvořen následovně: testované aplikace jsou v názvu reprezentovány číslem: `ldapsearch` (01, 02 a 03), `ldapadd` (04), `ldapcompare` (05), `ldapdelete` (06), `ldapexop` (07), `ldapmodify` (08), `ldapmodrdn` (09), `ldappasswd` (10), `ldapwhoami` (11) a `ldapurl` (12). Část testující komunikaci, při které klient ověřuje certifikát serveru, je označena jako *a* a část, při které klient ověřuje certifikát serveru a zároveň server ověřuje certifikát klienta je označena jako *b*. Jak část *a* tak i část *b*, každá z nich obsahuje 3 testové případy (nešifrovaná komunikace, komunikace přes TLS a SSL). Test aplikace `ldapurl` je specifický, protože nekomunikuje s OpenLDAP serverem, ale pro úplnost testů je také navržen. Test je označen pouze číslem 12 (není potřeba rozdělit test na části *a* a *b*).

Pro aplikaci ldapsearch jsou napsány 3 testy, které ověřují funkčnost komunikace s různým nastavením souboru `/etc/openldap/ldap.conf`. Test číslo 01 má cestu k certifikátu CA nastavenu pomocí parametru `TLS_CACERTDIR`, test 02 pomocí parametru `TLS_CACERT` a u testu 03 parametr TLS_CACERTDIR specifikuje cestu k NSS databázi, ve které je uložen

certifikát CA.

Test 01a

Popis testu: Testování `ldapsearch` s TLS/SSL s ověřením certifikátu serveru. Cesta k certifikátu CA je definována pomocí `TLS_CACERTDIR` v `ldap.conf`.
Příprava testu:

- Openldap server nastavíme pro použití s TLS a SSL. Server nakonfigurujeme v souboru `/etc/openldap/slapd.conf`. Musíme nastavit načtení potřebných schémat uložených v adresáři `/etc/openldap/schema/`. Nastavíme cesty k certifikátům pomocí `TLSCACertificateFile`, `TLSCertificateFile`, `TLSCertificateKeyFile`. Definujeme `rootdn` (dn pro navázání spojení) a `rootpw` (heslo pro přístup k LDAP adresáři v zašifrované podobě).
- Správně nastavíme hodnoty SLAPD_LDAP a SLAPD_LDAPS v souboru `/etc/sysconfig/ldap`. Hodnoty umožňují použití jak spojení `ldap://`, tak `ldaps://`.
- Nastavíme výchozí parametry v `/etc/openldap/ldap.conf`, které mohou využívat ldap klienti při svém běhu. Definujeme atributy `BASE` (výchozí DN) a `TLS_CACERTDIR` (specifikuje cestu k adresáři s certifikátem certifikační autority (CA)).

Akce testu:

- Vyhledáme záznamy v ldap pomocí `ldapsearch -H ldap://my-domain.com -x '*'` (parametr -H specifikuje URI ldap serveru, -x vynucuje použití jednoduché autentizace a * na konci příkazu uvádí hledaný výraz). Komunikace v tomto případě není šifrována.
- Vyhledáme záznamy v ldap pomocí `ldapsearch -H ldap://my-domain.com -x -ZZ '*'` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.
- Vyhledáme záznamy v ldap pomocí `ldapsearch -H ldaps://my-domain.com -x '*'` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldapsearch` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá data uložená v ldap serveru.

Test 01b

Popis testu: Testování `ldapsearch` s TLS/SSL s ověřením certifikátu serveru i klienta. Cesta k certifikátu CA je definována pomocí `TLS_CACERTDIR` v `ldap.conf`. Jediný rozdíl mezi testy 1a a 1b je v požadavku na ověření certifikátu klienta.
Příprava testu:

- Konfigurace všech souborů je stejná jako u testu číslo 1a. Pouze v souboru `slapd.conf` přidáme parametr `TLSVerifyClient` s hodnotou hard (vždy je vyžadováno ověření certifikátu klienta). Pro funkčnost tohoto testu je potřeba přidat certifikát klienta a klíč klienta do `/etc/openldap/cacerts/`.

- Požadujeme ověření certifikátu klienta, proto musí být v domovském adresáři uživatele vytvořen soubor `ldaprc`. Souborem ldaprc definujeme cesty k certifikátu a klíči klienta. Nastavení cest specifikujeme parametry `TLS_CERT` a `TLS_KEY`.

Akce testu: Fáze bude identická jako u testu číslo 1a.
Očekávaný výsledek: Příkaz `ldapsearch` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá data uložená v ldap serveru.

Test 02a

Popis testu: Testování `ldapsearch` s TLS/SSL s ověřením certifikátu serveru. Cesta k certifikátu CA je definována pomocí `TLS_CACERT` v `ldap.conf`. Jediný rozdíl mezi testy 1a a 2a je v parametru `TLS_CACERT` v `ldap.conf`.
Příprava testu: Fáze bude identická jako u testu číslo 1a, liší se pouze konfigurace souboru `ldap.conf`, kde je použit parametr `TLS_CACERT` (specifikuje cestu k certifikátu certifikační autority (CA)).
Akce testu: Fáze bude identická jako u testu číslo 1a.
Očekávaný výsledek: Příkaz `ldapsearch` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá data uložená v ldap serveru.

Test 02b

Popis testu: Testování `ldapsearch` s TLS/SSL s ověřením certifikátu serveru i klienta. Cesta k certifikátu CA je definována pomocí `TLS_CACERT` v `ldap.conf`. Jediný rozdíl mezi testy 2a a 2b je v požadavku na ověření certifikátu klienta.
Příprava testu: Fáze bude identická jako u testu číslo 1b. Pouze cesta k certifikátu CA je definována pomocí `TLS_CACERT` v `ldap.conf`.
Akce testu: Fáze bude identická jako u testu číslo 1a.
Očekávaný výsledek:Příkaz `ldapsearch` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá data uložená v ldap serveru.

Test 03a

Popis testu: Testování `ldapsearch` s TLS/SSL s ověřením certifikátu serveru. Certifikát CA uložen v NSS databázi. Jediný rozdíl mezi testy 1a a 3a je v hodnotě parametru `TLS_CACERTDIR` v `ldap.conf`.
Příprava testu: Fáze bude identická jako u testu číslo 1a, liší se pouze konfigurace souboru `ldap.conf`, kde parametr `TLS_CACERTDIR` má jinou hodnotu (specifikuje cestu k NSS databázi, ve které je uložen certifikát s CA).
Akce testu: Fáze bude identická jako u testu číslo 1a.
Očekávaný výsledek: Příkaz `ldapsearch` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá data uložená v ldap serveru.

Test 03b

Popis testu: Testování `ldapsearch` s TLS/SSL s ověřením certifikátu serveru i klienta. Certifikát CA uložen v NSS databázi. Jediný rozdíl mezi testy 3a a 3b je v požadavku na ověření certifikátu klienta.
Příprava testu: Fáze bude identická jako u testu číslo 1b. V přípravě jsou 2 změny. Certifikát CA uložen v NSS databázi a hodnotou parametru `TLS_CACERT` v `ldap.conf` je jméno klientského certifikátu, pod kterým byl uložen do databáze.
Akce testu: Fáze bude identická jako u testu číslo 1a.
Očekávaný výsledek: Příkaz `ldapsearch` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá data uložená v ldap serveru.

Test 04a

Popis testu: Testování `ldapadd` s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a.
Akce testu:

- Test přidání záznamu do ldap serveru pomocí `ldapadd -H ldap://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -f add1.ldif` (-D a -w definují dn a heslo, s těmito údaji klient navazuje spojení s ldap serverem, parametrem -f se specifikuje soubor s daty pro uložení). Komunikace v tomto případě není šifrována.
- Test přidání záznamu do ldap serveru pomocí `ldapadd -H ldap://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -f add2.ldif` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.
- Test přidání záznamu do ldap serveru pomocí `ldapadd -H ldaps://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -f add3.ldif` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldapadd` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a uloží zadaná data do ldap serveru.

Test 04b

Popis testu: Testování `ldapadd` s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 1b.
Akce testu: Fáze bude identická jako u testu číslo 4a.
Očekávaný výsledek: Příkaz `ldapadd` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a uloží zadaná data do ldap serveru.

Test 05a

Popis testu: Testování `ldapcompare` s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a.
Akce testu:

- Test na porovnání záznamu z ldap serveru na shodu se zadaným atributem a hodnotou. Porovnání se provádí pomocí `ldapcompare -H ldap://my-domain.com -x`

`'cn=Manager,dc=my-domain,dc=com' description:'Directory Manager'` (První hodnota zadává dn záznamu, ve kterém se má atribut hledat. Druhá hodnota definuje atribut a jeho hodnotu, se kterým se má porovnávat). Komunikace v tomto případě není šifrována.

- Test na porovnání záznamu z ldap serveru na shodu se zadaným atributem a hodnotou. Porovnání se provádí pomocí `ldapcompare -H ldap://my-domain.com -x -ZZ 'cn=Manager,dc=my-domain,dc=com' description:'Directory Manager'` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.
- Test na porovnání záznamu z ldap serveru na shodu se zadaným atributem a hodnotou. Porovnání se provádí pomocí `ldapcompare -H ldaps://my-domain.com -x 'cn=Manager,dc=my-domain,dc=com' description:'Directory Manager'` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldapcompare` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a jako výsledek vrátí TRUE (záznam cn=Manager,dc=my-domain,dc=com má atribut description a hodnotu 'Directory Manager').

Test 05b

Popis testu: Testování `ldapcompare` s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 1b.
Akce testu: Fáze bude identická jako u testu číslo 5a.
Očekávaný výsledek: Příkaz `ldapcompare` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a jako výsledek vrátí TRUE (záznam cn=Manager,dc=my-domain,dc=com má atribut description a hodnotu 'Directory Manager').

Test 06a

Popis testu: Testování `ldapdelete` s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a. Pouze soubor s daty `data.ldif` bude obsahovat další data, která se budou příkazem `ldapdelete` postupně mazat.
Akce testu:

- Test smazání záznamu z ldap serveru pomocí `ldapdelete -H ldap://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x 'cn=testusr1, dc=my-domain,dc=com'` (parametr na konci příkazu specifikuje záznam, který se má smazat). Komunikace v tomto případě není šifrována.
- Test smazání záznamu z ldap serveru pomocí `ldapdelete -H ldap://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -ZZ 'cn=testusr2, dc=my-domain, dc=com'` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.
- Test smazání záznamu z ldap serveru pomocí `ldapdelete -H ldaps://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x 'cn=testusr3, dc=my-domain,dc=com'` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldapdelete` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a smaže zadaná data z ldap serveru.

Test 06b

Popis testu: Testování `ldapdelete` s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 1b. Navíc budou do adresáře přidána dodatečná data, stejně jako u testu 6a.
Akce testu: Fáze bude identická jako u testu číslo 6a.
Očekávaný výsledek: Příkaz `ldapdelete` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a smaže zadaná data z ldap serveru.

Test 07a

Popis testu: Testování `ldapexop` s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a.
Akce testu:

- Test na funkčnost příkazu `ldapexop`, který provádí rozšiřujicí operace nad ldap. Pokud je příkaz spuštěn s parametrem whoami, bude výsledkem operace jméno s jakým klient navázal spojení se serverem. Rozšiřující dotaz se provádí pomocí `ldapexop -H ldap://my-domain.com -x whoami`. Komunikace v tomto případě není šifrována.
- Test na funkčnost příkazu `ldapexop`, který provádí rozšiřujicí operace nad ldap. Rozšiřující dotaz se provádí pomocí `ldapexop -H ldap://my-domain.com -x -ZZ whoami` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.
- Test na funkčnost příkazu `ldapexop`, který provádí rozšiřujicí operace nad ldap. Rozšiřující dotaz se provádí pomocí `ldapexop -H ldaps://my-domain.com -x whoami` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldapexop` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a jako výsledek vrátí jméno použité pro navázání spojení.

Test 07b

Popis testu: Testování `ldapexop` s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 1b.
Akce testu: Fáze bude identická jako u testu číslo 7a.
Očekávaný výsledek: Příkaz `ldapexop` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a jako výsledek vrátí jméno použité pro navázání spojení.

Test 08a

Popis testu: Testování `ldapmodify` s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 6a.
Akce testu:

- Test na funkčnost příkazu `ldapmodify`, který provádí modifikaci atributů a jejich hodnot u záznamu. Změna se provádí pomocí `ldapmodify -H ldap://my-domain.com`

`-D cn=Manager,dc=my-domain,dc=com -w x -f modify1.ldif` (parametr -f se specifikuje soubor s informacemi pro modifikaci). Komunikace v tomto případě není šifrována.

- Test na funkčnost příkazu `ldapmodify`, který provádí modifikaci atributů a jejich hodnot u záznamu. Změna se provádí pomocí `ldapmodify -H ldap://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -ZZ -f modify2.ldif` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.

- Test na funkčnost příkazu `ldapmodify`, který provádí modifikaci atributů a jejich hodnot u záznamu. Změna se provádí pomocí `ldapmodify -H ldaps://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -f modify3.ldif` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldapmodify` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a modifikuje zadané atributy záznamu.

Test 08b

Popis testu: Testování `ldapmodify` s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 6b.
Akce testu: Fáze bude identická jako u testu číslo 8a.
Očekávaný výsledek: Příkaz `ldapmodify` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a modifikuje zadané atributy záznamu.

Test 09a

Popis testu: Testování `ldapmodrdn` s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 6a.
Akce testu:

- Test na funkčnost příkazu `ldapmodrdn`, který provádí změnu relativního DN záznamu v ldap. Změna se provádí pomocí `ldapmodrdn -r -H ldap://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -f modrdn1.ldif` (parametr -r odstraní staré relativní DN ze záznamu). Komunikace v tomto případě není šifrována.

- Test na funkčnost příkazu `ldapmodrdn`, který provádí změnu relativního DN záznamu v ldap. Změna se provádí pomocí `ldapmodrdn -r -H ldap://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -ZZ -f modrdn2.ldif` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.

- Test na funkčnost příkazu `ldapmodrdn`, který provádí změnu relativního DN záznamu v ldap. Změna se provádí pomocí `ldapmodrdn -r -H ldaps://my-domain.com -D cn=Manager,dc=my-domain,dc=com -w x -f modrdn3.ldif` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldapmodrdn` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a změní relativní DN záznamu.

Test 09b

Popis testu: Testování `ldapmodrdn` s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 6b.
Akce testu: Fáze bude identická jako u testu číslo 9a.
Očekávaný výsledek: Příkaz `ldapmodrdn` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a změní relativní DN záznamu.

Test 10a

Popis testu: Testování `ldappasswd` s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 6a.
Akce testu:

- Test na funkčnost příkazu `ldappasswd`, který provádí změnu hesla ldap záznamu. Změna se provádí pomocí `ldappasswd -H ldap://my-domain.com -D cn=Manager, dc=my-domain,dc=com -w x -s 'password' 'cn=testusr1, dc=my-domain, dc=com'` (parametrem -s a jeho hodnotou se zadává nové heslo pro zadaný záznam). Komunikace v tomto případě není šifrována.

- Test na funkčnost příkazu `ldappasswd`, který provádí změnu hesla ldap záznamu. Změna se provádí pomocí `ldappasswd -H ldap://my-domain.com -D cn=Manager, dc=my-domain,dc=com -w x -ZZ -s 'password' 'cn=testusr2, dc=my-domain, dc=com'` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.

- Test na funkčnost příkazu `ldappasswd`, který provádí změnu hesla ldap záznamu. Změna se provádí pomocí `ldappasswd -H ldaps://my-domain.com -D cn=Manager, dc=my-domain,dc=com -w x -s 'password' 'cn=testusr3, dc=my-domain, dc=com'` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldappasswd` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a změní heslo pro daný ldap záznam.

Test 10b

Popis testu: Testování `ldappasswd` s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 6b.
Akce testu: Fáze bude identická jako u testu číslo 10a.
Očekávaný výsledek: Příkaz `ldappasswd` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a změní heslo pro daný ldap záznam.

Test 11a

Popis testu: Testování `ldapwhoami` s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a.
Akce testu:

- Test na funkčnost příkazu `ldapwhoami`, který zjistí jméno s jakým klient navázal spojení se serverem. Dotaz se provádí pomocí `ldapwhoami -H ldap://my-domain.com -x`. Komunikace v tomto případě není šifrována.
- Test na funkčnost příkazu `ldapwhoami`, který zjistí jméno s jakým klient navázal spojení se serverem. Dotaz se provádí pomocí `ldapwhoami -H ldap://my-domain.com -x -ZZ` (paramter -ZZ striktně vynucuje použití TLS). Komunikace přes TLS.
- Test na funkčnost příkazu `ldapwhoami`, který zjistí jméno s jakým klient navázal spojení se serverem. Dotaz se provádí pomocí `ldapwhoami -H ldaps://my-domain.com -x` (`ldaps://` u hodnoty parametru -H specifikuje použití SSL). Komunikace přes SSL.

Očekávaný výsledek: Příkaz `ldapwhoami` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a jako výsledek vrátí jméno použité pro navázání spojení.

Test 11b

Popis testu: Testování `ldapwhoami` s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 1b.
Akce testu: Fáze bude identická jako u testu číslo 11a.
Očekávaný výsledek: Příkaz `ldapwhoami` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a jako výsledek vrátí jméno použité pro navázání spojení.

Test 12

Popis testu: Testování `ldapurl`, který provádí formátování URL.
Příprava testu: U této funkce není nutná žádná přípravná fáze
Akce testu:

- Test na funkčnost příkazu `ldapurl`. Formátování se provádí pomocí `ldapurl -H ldap://my-domain.com '*'`. Formátování je provedeno pro nešifrovanou komunikaci.
- Test na funkčnost příkazu `ldapurl`. Formátování se provádí pomocí `ldapurl -H ldaps://my-domain.com '*'`. Formátování je provedeno pro šifrovanou komunikaci pomocí SSL.

Očekávaný výsledek: Příkaz `ldapurl` proběhne úspěšně pro oba příkazy a jako výsledek vrátí rozložené URL do jednotlivých parametrů.

4.2 Balíček krb5

Balíček kerberos podporuje šifrování komunikace mezi kerberos serverem a LDAP serverem pouze přes SSL. Nepodporuje ani ověření certifikátu klienta serverem při šifrovaném spojení. Z tohoto důvodu je pro tento balíček vytvořen jen jeden test (Test 1), který obsahuje 2 testové případy (nešifrovaná komunikace a komunikace přes SSL).
Samotný test komunikace je proveden požadavkem o vydání lístku u serveru pro distribuci lístků.

Test 1

Popis testu: Testování kerberos s SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a pro balíček openldap-clients.
Akce testu:

- Konfigurace balíčku se provádí v souboru `/etc/krb5.conf`. Konfigurací `dbmodules` definujeme použití LDAP jako části pro uložení dat. Musíme nastavit parametry pro projení s OpenLDAP serverem. Uvedením hodnoty kldap pro parametr `db_library` umožníme kerberos použití LDAP a dalšími parametry nastavíme údaje pro přístup. Definicí hodnoty `ldap_servers na ldap://my-domain.com` sdělíme kerberos, kde má hledat LDAP server.
- Test komunikace balíčku krb5 s OpenLDAP serverem bez šifrování. Test je proveden požadavkem o vydání lístku u serveru pro distribuci lístků.
- Nastavení Kerberos tak, aby komunikace mezi kerberos a ldap serverem byla šifrována přes SSL. Konfigurace bude stejná jako v předchozím případě bez šifrování, změní se pouze hodnota u parametru `ldap_servers` na `ldaps://my-domain.com`.
- Test komunikace balíčku krb5 s OpenLDAP serverem s šifrováním přes SSL. Test je proveden požadavkem o vydání lístku u serveru pro distribuci lístků.

Očekávaný výsledek: Získání lístku proběhne úspěšně pro oba způsoby nastavení kerberos (bez šifrování a komunikace přes SSL).

4.3 Balíček nss-pam-ldapd

Pro balíček nss-pam-ldapd jsou navrženy 2 testy. Test ověřující komunikaci, při které klient ověřuje certifikát serveru, je označen jako Test 1. Druhý test označený jako Test 2 ověřuje komunikaci, při které klient ověřuje certifikát serveru a zároveň server ověřuje certifikát klienta. Jak Test 1 tak i Test 2, každý z nich obsahuje 3 testové případy (nešifrovaná komunikace, komunikace přes TLS a SSL).
Test komunikace je proveden požadavkem getent passwd uživatel. Záznam uživatele je uložen v LDAP.

Test 1

Popis testu: Testování nss-pam-ldapd s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a pro balíček openldap-clients.
Akce testu:

- Konfigurace balíčku se provádí v souboru `/etc/nslcd.conf`. Potřebujeme nastavit parametry `uid`, `gid`, `uri` a `ssl` na hodnotu no pro nešifrovanou komunikaci.
  Testujeme funkčnost pomocí příkazu `getent passwd user1` (uživatel user1 je záznamem v LDAP serveru).
- Konfigurace nss-pam-ldapd s využitím TLS. Soubor `nslcd.conf` zůstane stejný jako v konfiguraci bez šifrování, ale změní se hodnota parametru `ssl`. Cestu k certifikátům nastavíme parametrem `tls_cacertdir`. `Tls_reqcert` s hodnotou demand zajistí, že se vždy použije TLS (komunikace nebude nikdy nešifrovaná).

Testujeme funkčnost pomocí příkazu `getent passwd user1` se šifrovanou komunikací přes TLS.

- Konfigurace nss-pam-ldapd s využitím SSL. Nastavení `nslcd.conf` se bude lišit v hodnotách parametrů `ssl` a `uri` v porovnání s nastavením pomocí TLS.
  Testujeme funkčnost pomocí příkazu `getent passwd user1` se šifrovanou komunikací přes SSL.

Očekávaný výsledek: Příkaz `getent passwd user1` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá požadované údaje o uživateli user1.

Test 2

Popis testu: Testování nss-pam-ldapd s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Konfigurace všech souborů je stejná jako u testu číslo 1. Pouze v souboru `slapd.conf` přidáme parametr `TLSVerifyClient` s hodnotou hard (vždy je vyžadováno ověření certifikátu klienta). Pro funkčnost tohoto testu je potřeba přidat certifikát klienta a klíč klienta do `/etc/openldap/cacerts/`.
Akce testu:

- Konfigurace balíčku se provádí v souboru `/etc/nslcd.conf`. Potřebujeme nastavit parametry `uid`, `gid`, `uri` a `ssl` na hodnotu no pro nešifrovanou komunikaci.
  Testujeme funkčnost pomocí příkazu `getent passwd user1` (uživatel user1 je záznamem v LDAP serveru).

- Konfigurace nss-pam-ldapd s využitím TLS. Soubor `nslcd.conf` zůstane stejný jako v konfiguraci bez šifrování, ale změní se hodnota parametru `ssl`. Cestu k certifikátům nastavíme parametrem `tls_cacertdir`. `Tls_reqcert` s hodnotou demand zajistí, že se vždy použije TLS (komunikace nebude nikdy nešifrovaná).
  Server vyžaduje ověření certifikátu klienta, a proto musí být vytvořen certifikát a klíč klienta v `/etc/openldap/cacerts/`. Cestu k certifikátům definujeme v konfiguračním souboru `nslcd.conf` parametry `tls_cert` a `tls_key`.
  Testujeme funkčnost pomocí příkazu `getent passwd user1` se šifrovanou komunikací přes TLS.

- Konfigurace nss-pam-ldapd s využitím SSL. Nastavení `nslcd.conf` se bude lišit v hodnotách parametrů `ssl` a `uri` v porovnání s nastavením komunikace přes TLS.
  Testujeme funkčnost pomocí příkazu `getent passwd user1` se šifrovanou komunikací přes SSL.

Očekávaný výsledek: Příkaz `getent passwd user1` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá požadované údaje o uživateli user1.

4.4 Balíček samba

Pro balíček samba jsou navrženy 2 testy. Test ověřující komunikaci, při které klient ověřuje certifikát serveru, je označen jako Test 1. Druhý test označený jako Test 2 ověřuje komunikaci, při které klient ověřuje certifikát serveru a zároveň server ověřuje certifikát klienta.

Jak Test 1 tak i Test 2, každý z nich obsahuje 3 testové případy (nešifrovaná komunikace, komunikace přes TLS a SSL).
Samotný test komunikace je proveden přístupem klienta projektu Samba k serveru Samba.

Test 1

Popis testu: Testování samba s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a pro balíček openldap-clients. Navíc je nutné do konfiguračního souboru serveru `slapd.conf` přidat parametr `index` s nastavením na `sambaSID,sambaPrimaryGroupSID,sambaDomainName eq` a načíst samba schéma.
Akce testu:

- Konfigurace balíčku se provádí v souboru `/etc/samba/smb.conf`. Parametr `passdb backend` s hodnotou `ldapsam:ldap://my-domain.com` specifikuje použití LDAP serveru pro uložení informací o uživatelích a skupinách. Paramter `ldap admin dn` uvádí dn, se kterým Samba navazuje spojení s LDAP serverem. `Ldap ssl` s hodnotou off znamená použití nešifrované komunikace mezi balíčkem samba a LDAP serverem. Test nešifrované komunikace mezi balíčkem samba a LDAP serverem pomocí přístupu uživatele smbtstuser (jeho záznam je v LDAP serveru) k Samba serveru pomocí `smbclient -L my-domain.com -U smbtstuser%heslo` (my-domain.com je doménové jméno, kde běží LDAP server).

- Konfigurace balíčku samba se šifrováním komunikace přes TLS je stejná jako v předcházejícím případě, změní se parametr `ldap ssl` na start tls. Test šifrované komunikace přes TLS mezi balíčkem samba a LDAP serverem pomocí přístupu uživatele smbtstuser.

- Konfigurace balíčku samba se šifrováním komunikace přes SSL je stejná jako jako v předcházejícím případě, změní se pouze parametry `ldap ssl` a `passdb backend`. Parametr `ldap ssl` bude mít hodnotu off a `passdb backend` hodnotu `ldapsam: ldaps://my-domain.com:636`. Test šifrované komunikace přes SSL mezi balíčkem samba a LDAP serverem pomocí přístupu uživatele smbtstuser.

Očekávaný výsledek: Příkaz `smbclient -L my-domain.com -U smbtstuser%heslo` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a uživatel se připojí k samba serveru.

Test 2

Popis testu: Testování samba s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Fáze bude identická jako u testu číslo 1b pro balíček openldap-clients. Navíc je nutné do konfiguračního souboru serveru `slapd.conf` přidat parametr `index` s nastavením na `sambaSID,sambaPrimaryGroupSID,sambaDomainName eq` a načíst samba schéma.
Akce testu: Fáze bude identická jako u testu číslo 1.
Očekávaný výsledek: Příkaz `smbclient -L my-domain.com -U smbtstuser%heslo` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a uživatel se připojí k samba serveru.

4.5 Balíček autofs

Pro balíček autofs jsou navrženy 2 testy. Test ověřující komunikaci, při které klient ověřuje certifikát serveru, je označen jako Test 1. Druhý test označený jako Test 2 ověřuje komunikaci, při které klient ověřuje certifikát serveru a zároveň server ověřuje certifikát klienta. Jak Test 1 tak i Test 2, každý z nich obsahuje 3 testové případy (nešifrovaná komunikace, komunikace přes TLS a SSL).
Test komunikace je proveden přístupem do adresáře, který se má automaticky připojovat.

Test 1

Popis testu: Testování autofs s TLS/SSL s ověřením certifikátu serveru.
Příprava testu: Fáze bude identická jako u testu číslo 1a pro balíček openldap-clients. Navíc je nutné do konfiguračního souboru serveru `slapd.conf` načíst autofs schéma. Musí být k dispozici zařízení, jež se bude připojovat. Aby zařízení bylo vždy dostupné, vytvoříme pro účely tohoto testu soubor, který připojíme k loop device.
Akce testu:

- Konfigurace balíčku autofs bez šifrování komunikace. Konfigurace se provádí v souboru `/etc/sysconfig/autofs`. Šifrování definujeme v souboru `/etc/autofs_ldap_auth.conf`. URI LDAP serveru definuje atribut `LDAP_URI` s hodnotou `ldap://my-domain.com`.
  Nastavení šifrování komunikace definuje soubor `autofs_ldap_auth.conf`. Nastavíme použití nešifrované komunikace pomocí `usetls` a `tlsrequired` s hodnotami no.
  Ověříme funkčnost komunikace mezi balíčkem autofs a OpenLDAP serverem pomocí příkazu `ls /misc/loop` (s pomocí záznamů v LDAP by se mělo loop device připojit do adresáře `/misc`).

- Konfigurace balíčku autofs se šifrováním komunikace přes TLS. Soubor autofs nastavíme stejně jako v předchozím případě. Změní se pouze parametry `usetls` a `tlsrequired` v souboru `autofs_ldap_auth.conf`. `Usetls` i `tlsrequired` budou mít hodnoty yes. Ověříme funkčnost komunikace pomocí příkazu `ls /misc/loop`.

- Konfigurace balíčku autofs se šifrováním komunikace přes SSL. Autofs má stejné parametry jako v předchozích případech, pouze parametr `ldap uri` změní hodnotu na `ldaps://my-domain.com`. Parametry `usetls` a `tlsrequired` v `autofs_ldap_auth.conf` mají hodnotu no. Ověříme funkčnost komunikace pomocí příkazu `ls /misc/loop`.

Očekávaný výsledek: Příkaz `ls /misc/loop` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá údaje o souborech dostupných v připojeném adresáři /misc/loop.

Test 2

Popis testu: Testování autofs s TLS/SSL s ověřením certifikátu serveru i klienta.
Příprava testu: Fáze bude identická jako u testu číslo 1b pro balíček openldap-clients. Navíc je nutné do konfiguračního souboru serveru `slapd.conf` načíst autofs schéma. Musí být k dispozici zařízení, jež se bude připojovat. Aby zařízení bylo vždy dostupné, vytvoříme pro účely tohoto testu soubor, který připojíme k loop device.
Akce testu: Fáze bude identická jako u testu číslo 1.

Očekávaný výsledek: Příkaz `ls /misc/loop` proběhne úspěšně (pro všechny 3 způsoby komunikace - bez šifrování, komunikace přes TLS, nebo SSL) a získá údaje o souborech dostupných v namountovaném adresáři /misc/loop.

Kapitola 5

Automatizace testů, jejich implementace a pokrytí

Testy měly být podle zadání automatizovaně testovány pomocí systému AutoQA. AutoQA se stále vyvíjí a v současné době není vývoj tak daleko, aby si uživatel mohl navrhnout automatické testování jakéhokoli balíčku. Libovolné testování není možné z několika důvodů. Prvním důvodem jsou neexistující spouštěče (anglicky trigger), které by zaznamenaly aktualizaci balíčku. Testy by nemohly být spouštěny při vydání každého nového balíčku, ale jen při sestavování nové verze distribuce (například RC (Release Candidate) verze). Zabudování libolných testů tak, aby se spuštěly v době sestování nové verze distribuce není nyní prioritou Red Hat. Dalším problémem je neexistence čistého systému před každým testem, což znamená problém při destruktivních testech. Řešením by bylo použití zálohy stavu virtuálního systému (anglicky snapshot), ale řešení není nyní realizováno.
Problém nebrání automatizaci samotných testů, protože AutoQA jen spouští již vytvořené testy při výskytu definované události. Automatizace testů pomocí AutoQA bude popsána obecně.

V kapitole bude popsáno jak lze automatizovat testy pro systém AutoQA a bude popsána knihovna BeakerLib pro tvorbu testovacích skriptů pro Bash. Dále bude zmíněno na jakých distribucích byly testy provedeny. V závěru kapitoly je zhodnoceno pokrytí a navrženo vylepšení pokrytí testů.

5.1 Automatizace testů pro testovací systém AutoQA

Test pro AutoQA se skládá z několika částí. Musí být vytvořen nový adresář, ve kterém budou testy a řídicí soubory AutoQA. Jméno nového adresáře bude zároveň použito i jako jméno testu. Řídicí soubor s názvem control definuje metadata testu jako jméno autora, délku testu (short, medium, long), typ testu (klient nebo server), třídu a kategorii (funkcionální, zátěžové, výkonové a regresní) testu. Dalším řídicím souborem je control.autoqa, definující kdy má být test spuštěn, pro jakou architekturu a podobně. Mohou být definovány i požadavky na test, například spuštění na virtuálním stroji (destruktivní test). Třetí řídicí soubor definuje objekt reprezentující test. Soubor je napsán v jazyce python a má název adresáře s příponou py. Testovací objekt připravuje podmínky vhodné pro test, spouští testovací kód a shromažďuje výsledky. Testy by měly vracet smysluplný návratový kód.
Všechny navržené testy jsou implementovány jako skripty pro Bash s využitím knihovny BeakerLib. BeakerLib usnadňuje psaní testů a jednoduše vrací smysluplný návrotvý kód.

5.1.1 Automatizace testů pro Bash

Navržené testy jsou implementovány jako skripty pro Bash s využitím knihovny BeakerLib. Pro každý testovaný balíček je vytvořen jeden skript nazvaný `runtest.sh` a je doplněn souborem `PURPOSE` s popisem daného testu.

BeakerLib

BeakerLib je knihovna pro usnadnění tvorby testovacích skriptů pro Bash. Test zaznamenává jako žurnálový soubor, který může vypsat v přehledném textovém nebo xml formátu. Disponuje logickým rozvržením testových případů přímo ve skriptu. Skript může obsahovat fáze nastavení (`rlPhaseStartSetup`), provedení testu (`rlPhaseStartTest`) a úklid po testu (`rlPhaseStartCleanup`). BeakerLib poskytuje nástroje usnadňující kontrolu existence balíčků, spouštění, zastavení či obnovu služeb do původního stavu po ukončení testu. Rovněž umožňuje zálohu souborů a jejich obnovu pomocí jednoho příkazu na konci testu. Podobné usnadnění poskytuje i při porovnávání hodnot. Snadno lze zkontrolovat existenci souboru, výrazu v souboru nebo porovnání shody dvou souborů. Nejpoužívanějším příkazem knihovny je `rlRun` sloužící pro spuštění příkazu, u kteréhoje možnost zadat očekávanou správnou návratovou hodnotu. Podle návratové hodnoty se vyhodnotí výsledek `rlRun` jako PASS nebo FAIL. U příkazu lze uvést komentář, který se vyskytuje ve výpisu o průběhu testu. Každá fáze testového případu vypisuje celkový výsledek fáze – pass nebo fail. Celá fáze skončí fail, pokud některý z příkazů ve fázi skončí neúspěchem.

Implementace testů s využitím BeakerLib

Testy navržené ve 4. kapitole Plán testování jsou implementovány jako skripty pro Bash a využívají možností knihovny BeakerLib. Na rozdíl od navržených testů jsou implementované testy doplněny na začátku o fázi instalace závislostí a o fázi zálohy konfigurace služeb, klienta a serveru. Je provedena také záloha stavu samotných služeb. Na závěr jsou testy doplněny o fázi testující SELinux a o fázi vracející systém do stavu před testem.

Před začátkem každého testu se provede instalace závislostí. Pro každý test existuje množina balíčků, o kterých se předpokládá, že jsou nainstalovány (závislosti). To však nemusí v testovaném systému platit a proto je potřeba před testem zkontrolovat jestli žádná závislost nechybí a případně ji doinstalovat. Závislosti jsou před každým testem uvedeny v poli `PACKAGES` a zkontrolvány s využitím funkce `rlCheckRpm` z knihovny BeakerLib. Následně je provedeno zastavení a uložení stavu testovaných služeb s pomocí funkce `rlServiceStop`. Zálohovány jsou také konfigurace jednotlivých služeb a serveru pomocí funkce `rlFileBackup`.

V další části jsou implementovány fáze testů navržené v kapitole 4. Knihovna BeakerLib usnadňuje tvorbu skriptu i v těchto fázích a nejvyužívanější funkcí je `rlRun`, která provede zadaný příkaz, porovná jeho návratovou hodnotu s očekávanou a do výpisu testu přidá komentář funkce i s výsledkem (úspěch nebo neúspěch).

Další fáze obsahuje SELinux test, jež kontroluje zda nedošlo během testu k problémům se SELinux. Kontrola je provedena pomocí příkazu `ausearch -m AVC -ts recent`. Případný nalezený problém je nutné zkontrolovat, protože parametr `-ts recent` vypíše problémy za posledních 10 minut a test může být ve vyjímečných případech ovlivněn prostředím.

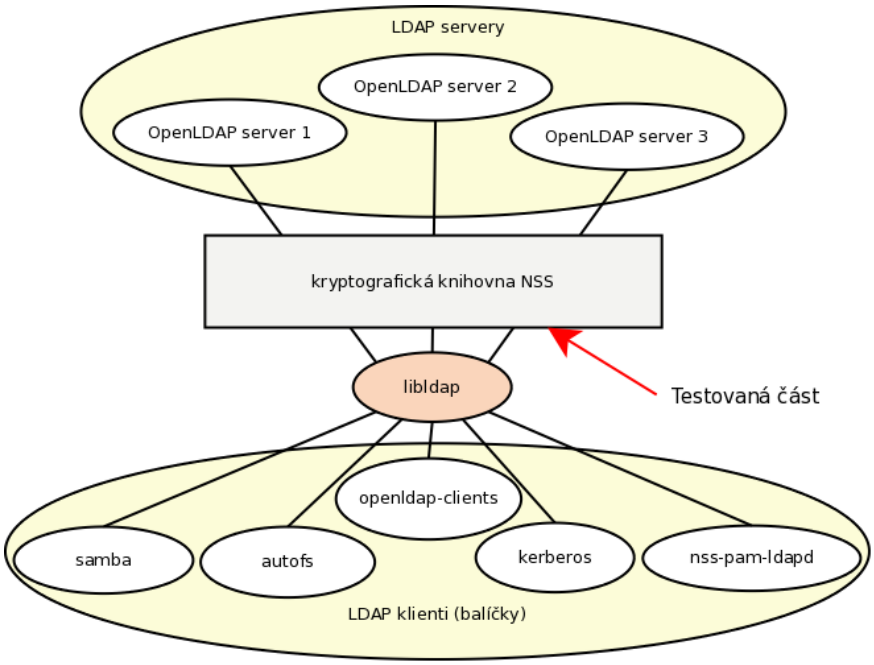
Na závěr testu je provedena fáze uklizení systému. Jsou obnoveny zálohované konfigurace pomocí jediné funkce `rlFileRestore`. Obnoven je i stav zálohovaných služeb s využitím funkce `rlServiceRestore`.

5.2 Provedení testů na distribuci Fedora nebo Red Hat Enterprise Linux

Všechny testy byly vykonány na distribuci Fedora i RHEL. Pouze test pro balíček samba nemohl být proveden na distribuci RHEL, protože ve fázi přípravy testu používá balíček smbldap-tools, který není v distribuci RHEL dostupný.
Aktuální nejnovější verze Fedory je 14, ale testování balíčků této verze není pro Red Hat prioritou. Proto byly testovány především balíčky z verze 15 a Rawhide (poslední vývojová verze, aktuálně 16). Testy na distribuci RHEL byly provedeny spolupracujícím kolegou z firmy Red Hat, Ondřejem Morišem. Systém RHEL není volně dostupný, proto nebyly testy vykonány přímo autorem této práce.

5.3 Zhodnocení pokrytí testů

Jednotlivé balíčky dokážou komunikovat s LDAP serverem přes knihovnu libldap jen jedním způsobem. Komunikace mezi klienty a serverem je znázorněna na obrázku 5.1. Případy užití podle manuálových stránek k ldap (`man ldap`) pokrývají testování šifrované komunikace s využitím certifikátů všemi podporavanými způsoby. Podporovanými způsoby jsou ověření certifikátu serveru klientem nebo ověření certifikátu klienta serverem a zároveň ověření certifikátu serveru klientem při šifrované komunikaci přes SSL nebo TLS.
Většina z testovaných balíčků může použít všechny tyto možnosti. Výjimkou je služba kerberos, která umí pouze ověření certifikátu serveru klientem při šifrované komunikaci přes SSL.



Obrázek 5.1: Komunikace mezi klienty a serverem

5.4 Návrh oblastí testování OpenLDAP vedoucí ke zlepšení pokrytí testů

Navržené testy pokrývají možnosti šifrování jednotlivých balíčků všemi dostupnými způsoby. Tato oblast nenabízí více možností, jak navržené testy rozšířit o další možnosti. Distribuce Fedora obsahuje mnoho dalších balíčků, které dokážou spolupracovat s LDAP serverem. Přehled balíčků, jež mohou využívat LDAP, je vypsán na stránce [1]. Rozšíření případů užití i na tyto balíčky vede ke zlepšení pokrytí testů.

Kapitola 6

Závěr

Cílem práce bylo vytvořit sadu testů pro OpenLDAP/NSS. Jednotlivé testy se zaměřují na ověření funkčnosti šifrovaného spojení mezi LDAP klienty (balíčky) a OpenLDAP serverem. Při šifrovaném spojení se testuje ověření certifikátu serveru klientem a pokud testovaný balíček podporuje i ověření certifikátu klienta serverem, je testována i tato funkčnost.
V rámci bakalářské práce bylo vytvořeno pět automatizovaných testů pro Bash s využitím knihovny BeakerLib. Každý z testů ověřuje správnou funkčnost komunikace vybraných balíčků, kterými jsou openldap-clients, samba, autofs, krb5 a nss-pam-ldapd. S vyjímkou balíčku krb5 všechny testy ověřují funkčnost šifrované komunikace přes TLS i SSL a testují možnost ověření certifikátu klienta serverem. Balíček krb5 podporuje pouze šifrované spojení pomocí SSL a neumožňuje ověřit certifikát klienta serverem. Všechny testy byly provedeny na distribuci Fedora ve verzích 14, 15 a Rawhide a na distribuci RHEL. Vyjímkou je test pro balíček samba, jež nemohl být proveden na distribuci RHEL. Důvodem je nedostupnost balíčku smbldap-tools v distribuci RHEL, který skript používá ve fázi přípravy testu.

Provedení testů odhalilo mnoho chyb ve funkčnosti šifrovaného spojení, které jsou detailněji popsány v podkapitole 6.1. Nalezené problémy byly zaznamenány do systému Red Hat Bugzilla pro oznamování chyb. Testy jsou k dispozici firmě Red Hat a mohou být opakovaně spuštěny při vydání nových verzí balíčků, aby se zkontrolovala správná funkčnost šifrování i v nové verzi. Jestliže systém AutoQA pokročí ve vývoji, mohou být testy začleněny do tohoto systému a automaticky spouštěny při vydání nové verze balíčku.
Do budoucna by sada testů mohla být rozšířena i na další balíčky, které dokáží spolupracovat s OpenLDAP serverem. Seznam balíčků spolupracujících s LDAP lze najít na internetové stránce [1].

6.1 Nalezené problémy

Po implementaci testů a jejich spuštěním na distribucích Fedora a RHEL bylo objeveno mnoho problémů. Některé jsou zaznamenány jako chyby, jiné zde jen zmíním. Určité potíže se vyskytly jen pro balíčky z distribuce Fedora verze 14. Zanedlouho vyjde verze Fedora 15, v níž jsou některé zmíněné problémy již opraveny a není nutné je zaznamenávat jako chyby.

Pro aktuální vývojovou verzi distribuce Fedora Rawhide bylo nalezeno několik problémů. Funkce ldapexop z balíčku openldap-clients skončí s chybou v jedné z knihoven a je zaznamenána jako chyba číslo 699683. Samba klient nedokáže šifrovaně komunikovat

se serverem, pokud LDAP server vyžaduje ověření certifikátu klienta. Chyba byla zadána s číslem 695447. Autofs neumí komunikovat šifrovaně přes TLS nebo SSL s LDAP a problém je zaregistrován jako chyba číslo 695142. Problém byl nalezen i v balíčku nss-pam-ldapd. SELinux zabrání službě `nslcd` v operaci sys_nice a chyba je nahlášena s číslem 701587.

V distribuci RHEL verze 6.1 se také objevily problémy. SELinux zabrání službě `kadmin` při testu balíčku kerberos v operaci setsched. Chyba byla oznámena s číslem 698923. Autofs nedokáže šifrovaně komunikovat se serverem, pokud LDAP server vyžaduje ověření certifikátu klienta. Problém byl zaregistrován jako chyba číslo 695141. Funkce ldapadd z balíčku openldap-clients skončí s chybou se specifickými daty ve formátu ldif, u nichž chybí nový řádek na konci souboru. Chyba je zaznamenána s číslem 698921.

Na verzi Fedora 14 byly pro srovnání vykonány stejné testy jako na verzi Fedora Rawhide. Klient projektu Samba nedokáže vůbec šifrovaně komunikovat se serverem. Autofs nedokáže šifrovaně komunikovat se serverem, pokud LDAP server vyžaduje ověření certifikátu klienta. SELinux zabrání službě `kadmin` při testu balíčku kerberos v operaci setsched. Kerberos nedokáže ani šifrovaně komunikovat s LDAP serverem pomocí SSL. Baličeк nss-pam-ldapd umožňí šifrované spojení jen v debug módu, ve standardním módu dojde k chybě. Tento problém je zaznamenán jako chyba číslo 688771.

Literatura

[1] Test Day:2010-10-14 OpenLDAP/NSS [online]. https://fedoraproject.org/wiki/Test_Day:2010-10-14_OpenLDAP/NSS#Packages_list, 16.9.2010 [cit. 2011-04-07].

[2] Software Testing Automation Framework (STAF) [online]. http://staf.sourceforge.net/, 1998, last modified on April 01 2011 [cit. 2011-04-06].

[3] nss-pam-ldapd: NSS and PAM modules for lookups using LDAP [online]. http://arthurdejong.org/nss-pam-ldapd/, 2006 [cit. 2011-04-07].

[4] Kerberos (protocol). In Wikipedia : the free encyclopedia [online]. http://en.wikipedia.org/wiki/Kerberos_protocol, 25.2.2002, last modified on 8.4.2011 [cit. 2011-04-19].

[5] Software testing. In Wikipedia : the free encyclopedia [online]. http://en.wikipedia.org/wiki/Category:Software_testing, 5.12.2001, last modified on 6.4.2011 [cit. 2011-04-06].

[6] Network Security Services (NSS) [online]. http://www.mozilla.org/projects/security/pki/nss/, c1998-2011 [cit. 2011-01-30].

[7] Automatické připojování svazků [online]. http://www.penguin.cz/řadek/book/unix/autofs.html, c2004-2010 [cit. 2011-04-07].

[8] Phoronix Test Suite [online]. http://www.phoronix-test-suite.com, c2008 - 2011 [cit. 2011-04-06].

[9] Introduction to SSL [online]. https://developer.mozilla.org/en/Introduction_to_SSL, c2011 [cit. 2011-01-30].

[10] AutoQA [online]. http://fedoraproject.org/wiki/AutoQA, c2011, last modified on 28 March 2011 [cit. 2011-04-06].

[11] AutoQA Use Cases [online]. http://fedoraproject.org/wiki/AutoQA_Use_Cases, c2011, last modified on 28 March 2011 [cit. 2011-04-06].

[12] AutoQA architecture [online]. http://fedoraproject.org/wiki/AutoQA_architecture, c2011, last modified on 30 March 2011 [cit. 2011-04-06].

[13] MATOUŠEK, P.: *Adresářové služby : Studijní opora k předmětu ISA*. Brno: Fakulta informačních technologií VUT, 2010, 14 s.

[14] MYERS, G. J.; Sandler, C.: *The Art of Software Testing*. John Wiley & Sons, druhé vydání, 2004, ISBN 0471469122.

Dodatek A

Konfigrační soubory

V této části přílohy budou uvedeny příklady konfiguračních souborů důležitých částí testů. Uvedeny budou konfigurační soubory OpenLDAP serveru, knihovny OpenLDAP a důležitých částí kerberos, autofs, samba a nss-pam-ldapd.

A.1 Soubory OpenLDAP serveru

Nastavení OpenLDAP serveru se provádí souborem `/etc/openldap/slapd.conf`. Důležité je načtení potřebných schémat pro data v LDAP pomocí příkazu `include`. Pro funkčnost šifrování se musí nastavit parametry `TLSCipherSuite`, `TLSCACertificateFile`, `TLSCertificateFile` a `TLSCertificateKeyFile`. Jestliž je vyžadováno ověření certifikátu klienta serverem, bude v nastavení přidán parametr `TLSVerifyClient demand`.

Soubor /etc/openldap/slapd.conf

```
include         /etc/openldap/schema/core.schema
include         /etc/openldap/schema/cosine.schema
include         /etc/openldap/schema/inetorgperson.schema
include         /etc/openldap/schema/nis.schema

allow bind_v2

loglevel 1

pidfile         /var/run/openldap/slapd.pid
argsfile        /var/run/openldap/slapd.args

TLSCipherSuite        HIGH:MEDIUM:+SSLv2
TLSCACertificateFile  /etc/openldap/cacerts/cacert.pem
TLSCertificateFile    /etc/openldap/cacerts/servercrt.pem
TLSCertificateKeyFile /etc/openldap/cacerts/serverkey.pem

database        bdb
suffix          ,,dc=my-domain,dc=com‘‘
rootdn          ,,cn=Manager,dc=my-domain,dc=com‘‘
rootpw          {SSHA}tOSmeQCcYIm1S9ujgpg2Km5rpUnR9dRB
```

```
directory       /var/lib/ldap/

index objectClass                       eq,pres
index ou,cn,mail,surname,givenname      eq,pres,sub
index uidNumber,gidNumber,loginShell    eq,pres
index uid,memberUid                     eq,pres,sub
index nisMapName,nisMapEntry            eq,pres,sub
```

A.2 Nastavení libldap (knihovna openldap)

Nastavení důvěryhodné certifikační autority (CA) je provedeno v souboru `/etc/openldap/ldap.conf`. Při požadavku na ověření certifikátu klienta je nutné vytvořit soubor `ldaprc` v domovském adresáři uživate. Soubor `ldaprc` obsahuje cesty k certifikátu a klíči klienta.

Nastavení ldap.conf s cestou k adresáři s CA

```
BASE dc=my-domain,dc=com
BINDDN dc=Manager,dc=my-domain,dc=com
BINDPW x
TLS_CACERTDIR /etc/openldap/cacerts/
```

Nastavení ldap.conf s cestou k CA

```
BASE dc=my-domain,dc=com
BINDDN dc=Manager,dc=my-domain,dc=com
BINDPW x
TLS_CACERT /etc/openldap/cacerts/cacert.pem
```

Nastavení ldap.conf, CA je v nss databázi

```
BASE dc=my-domain,dc=com
BINDDN dc=Manager,dc=my-domain,dc=com
BINDPW x
TLS_CACERTDIR /tmp/openldap-clients-nssdb/
```

Nastavení ldaprc s cestou k certifikátu klienta

```
TLS_CERT        /etc/openldap/cacerts/clientcrt.pem
TLS_KEY         /etc/openldap/cacerts/clientkey.pem
TLS_REQCERT     demand
```

Nastavení ldaprc, certifikát klienta je v nss databázi

```
TLS_CERT        Client cert
TLS_REQCERT     demand
```

A.3 Konfigurační soubor kerberos - /etc/krb5.conf

Balíček kerberos se nastavuje v souboru `/etc/krb5.conf`. Soubor nastavuje parametry autentizačního systému Kerberos. V části `[dbmodules]` se nastavují parametry pro projení s OpenLDAP serverem. Pro nastavení šifrovaného spojení s OpenLDAP serverem pomocí SSL je nutné změnit parametr `ldap_servers` v části `[dbmodules]` a nastavit jeho hodnotu na `ldaps://my-domain.com`.
Pro správné propojení Kerberos s OpenLDAP musí být do LDAP adresáře uložena potřebná data. Důležitý je především záznam pro kerberos s třídou `objectClass krbContainer`. Aby mohl být přidán záznam s touto třídou, musí být v nastavení OpenLDAP serveru načteno schéma pro Kerberos.

Nastavení kerberos bez šifrování

```
[logging]
 default = FILE:/var/log/krb5libs.log
 kdc = FILE:/var/log/krb5kdc.log
 admin_server = FILE:/var/log/kadmind.log

[libdefaults]
 default_realm = MY-DOMAIN.COM
 dns_lookup_realm = false
 dns_lookup_kdc = false

[realms]
 MY-DOMAIN.COM = {
  kdc = my-domain.com:88
  admin_server = my-domain.com:749
 }

[domain_realm]
 .my-domain.com = MY-DOMAIN.COM
 my-domain.com = MY-DOMAIN.COM

[kdc]
 profile = /var/kerberos/krb5kdc/kdc.conf

[appdefaults]
 pam = {
   debug = false
   ticket_lifetime = 36000
   renew_lifetime = 36000
   forwardable = true
   krb4_convert = false
 }

[dbmodules]
  MY-DOMAIN.COM = {
```

```
  db_library = kldap
  ldap_kerberos_container_dn = ,,cn=Kerberos,dc=my-domain,dc=com‘‘
  ldap_kdc_dn = ,,cn=Manager,dc=my-domain,dc=com‘‘
  ldap_kadmind_dn = ,,cn=Manager,dc=my-domain,dc=com‘‘
  ldap_service_password_file = /var/kerberos/krb5kdc/ldap_passwd
  ldap_servers = ldap://my-domain.com
 }
```

Soubor s daty pro ldap ve formátu ldif

```
dn: dc=my-domain,dc=com
objectClass: dcObject
objectClass: organization
dc: my-domain
o: my-domain
description: my-domain

dn: cn=Manager,dc=my-domain,dc=com
objectClass: organizationalRole
cn: Manager
description: Directory Manager

dn: cn=Kerberos,dc=my-domain,dc=com
objectClass: krbContainer
cn: Kerberos
```

A.4 Konfigurační soubory autofs

Pro propojení autofs s OpenLDAP serverem je nutné nastavit, aby autofs hledal informace potřebné pro svou činnost v LDAP adresáři. Nastavení je provedeno v souboru `/etc/nsswitch.conf` parametrem `automount ldap`. Do LDAP adresáře musí být uložena potřebná data. Aby mohly být přidány záznamy pro autofs se specifickými třídami `automountMap` a `automount`, musí být v nastavení OpenLDAP serveru načteno schéma pro autofs. Záznamy `ou=auto.master,dc=my-domain,dc=com` a `ou=auto.misc,dc=my-domain, dc=com` nahrazují nastavení autofs, standardně uložené v lokálních souborech `/etc/auto.master` a `/etc/auto.misc`. Záznam `cn=/misc,ou=auto.master,dc=my-domain, dc=com` specifikuje kde se mají hledat informace pro připojování do adresáře `/misc`. Záznamem `cn=loop,ou=auto.misc,dc=my-domain,dc=com` definujeme co se bude připojovat do adresáře `/misc`.

Služba `autofs` se nastavuje v souboru `/etc/sysconfig/autofs`. V souboru je nastaveno URI OpenLDAP serveru, na který se má připojit. Pro použití automount map Parametry `MAP_OBJECT_CLASS`, `ENTRY_OBJECT_CLASS`, `MAP_ATTRIBUTE`, `VALUE_ATTRIBUTE` nastavují použití autofs schématu, které je nutné pro uložení informací o připojovaných adresářích. Nastavení šifrování spojení mezi autofs a OpenLDAP serverm se specifikuje v souboru `/etc/autofs_ldap_auth.conf`.

Soubor /etc/nsswitch.conf pro použití autofs s ldap

```
passwd:     files ldap
shadow:     files
group:      files ldap
hosts:      files
automount:  ldap
services:   files
```

Soubor s daty pro ldap ve formátu ldif

```
dn: dc=my-domain,dc=com
objectClass: dcObject
objectClass: organization
dc: my-domain
o: my-domain
description: my-domain

dn: cn=Manager,dc=my-domain,dc=com
objectClass: organizationalRole
cn: Manager
description: Directory Manager

dn: ou=auto.master,dc=my-domain,dc=com
objectClass: top
objectClass: automountMap
ou: auto.master

dn: ou=auto.misc,dc=my-domain,dc=com
objectClass: top
objectClass: automountMap
ou: auto.misc

dn: cn=/misc,ou=auto.master,dc=my-domain,dc=com
objectClass: automount
automountInformation:
ldap://my-domain.com:ou=auto.misc,dc=my-domain,dc=com --timeout 60
cn: /misc

dn: cn=loop,ou=auto.misc,dc=my-domain,dc=com
objectClass: automount
automountInformation: -fstype=ext4 :/dev/loop0
cn: loop
```

Konfigurace autofs s TLS

Nastavení souboru /etc/sysconfig/autofs:

```
TIMEOUT=300
```

```
BROWSE_MODE=,,no‘‘

SEARCH_BASE=,,dc=my-domain,dc=com‘‘

LDAP_URI=,,ldap://my-domain.com‘‘

MAP_OBJECT_CLASS=,,automountMap‘‘
ENTRY_OBJECT_CLASS=,,automount‘‘
MAP_ATTRIBUTE=,,automountMapName‘‘
VALUE_ATTRIBUTE=,,automountInformation‘‘

USE_MISC_DEVICE=,,yes‘‘
OPTIONS=,, -d -v ‘‘
```

Nastavení TLS šifrování v `/etc/autofs_ldap_auth.conf`:

```
<?xml version=,,1.0‘‘ ?>
<autofs_ldap_sasl_conf
usetls=,,yes‘‘
tlsrequired=,,yes‘‘
/>
```

A.5 Konfigurační soubory služby Samba

Nastavení parametrů služby Samba se provádí v souboru `/etc/samba/smb.conf`. Pro propojení služby s OpenLDAP serverem je nutné nastavit parametry spojení. Parametr `passdb backend` definuje URI OpenLDAP serveru a parametr `dap ssl` nastavuje použití šifrovaného spojení. Parametr `add user script` definuje použití nástroje `smbldap-useradd` z balíčku smbldap-tools pro přidání záznamu uživatele.

Konfigurace služby Samba s TLS v /etc/samba/smb.conf

```
[global]
workgroup = LDAP
netbios name = LDAPSERVER
enable privileges = yes
server string = SAMBA-LDAP Server

ldap passwd sync = yes
passdb backend = ldapsam:ldap://my-domain.com
ldap admin dn = ,,cn=Manager,dc=my-domain,dc=com‘‘
ldap ssl = start tls
ldap delete dn = no
ldap user suffix = ou=People
ldap suffix = ,,dc=my-domain,dc=com‘‘

add user script = /usr/sbin/smbldap-useradd -m ,,%u‘‘
```

Soubor s daty pro ldap ve formátu ldif

```
dn: dc=my-domain,dc=com
objectClass: dcObject
objectClass: organization
dc: my-domain
o: my-domain
description: my-domain

dn: cn=Manager,dc=my-domain,dc=com
objectClass: organizationalRole
cn: Manager
description: Directory Manager

# Setting up container for Users OU
dn: ou=People,dc=my-domain,dc=com
objectclass: top
objectclass: organizationalUnit
ou: People

# Setting up admin handle for People OU
dn: cn=admin,ou=People,dc=my-domain,dc=com
cn: admin
objectclass: top
objectclass: organizationalRole
objectclass: simpleSecurityObject
userPassword: {SSHA}Oc8YC4fJW9skMc5myDcnX16hTBBnLBc0

# LDAP, my-domain.com
dn: sambaDomainName=LDAP,dc=my-domain,dc=com
objectClass: top
objectClass: sambaDomain
objectClass: sambaUnixIdPool
sambaDomainName: LDAP
sambaSID: S-1-5-21-2847062194-4106847480-171505460
gidNumber: 1000
sambaNextRid: 1000
uidNumber: 1002
```

Konfigurace smbaldap-tools

Balíček smbaldap-tools je použit pro přidání záznamu uživatele do LDAP. Je nutné nastavit parametr `SID` specifikující jednoznačné identifikační číslo domény.
Parametr `sambaUnixIdPooldn` definuje oblast hledání Samba záznamů v LDAP serveru. Soubor `/etc/smbldap-tools/smbldap.conf`:

```
SID=,,S-1-5-21-2847062194-4106847480-171505460‘‘
sambaDomain=,,LDAP‘‘
ldapTLS=,,0‘‘
```

```
suffix=,,dc=my-domain,dc=com‘‘
sambaUnixIdPooldn=,,sambaDomainName=LDAP,${suffix}‘‘
scope=,,sub‘‘
defaultUserGid=,,513‘‘
defaultComputerGid=,,550‘‘
userGecos=,,%U‘‘
userLoginShell=,,/bin/bash‘‘
userHome=,,/home/%U‘‘
skeletonDir=,,/etc/skel‘‘
```

Soubor /etc/smbldap-tools/smbldap bind.conf:

```
slaveDN=,,cn=Manager,dc=my-domain,dc=com‘‘
slavePw=,,x‘‘
masterDN=,,cn=Manager,dc=my-domain,dc=com‘‘
masterPw=,,x‘‘
```

A.6 Konfigurační soubor nss-pam-ldapd - /etc/nslcd.conf

V nastavení nss-pam-ldapd se musí zadat parametry s URI LDAP serveru, nastavení šifrování spojení a cestu k certifikátu CA.

Nastavení nslcd s TLS

```
uid nslcd
gid ldap
uri ldap://my-domain.com
binddn cn=Manager,dc=my-domain,dc=com
bindpw x
ssl start_tls
tls_reqcert demand
tls_cacertdir /etc/openldap/cacerts/
```

Dodatek B

Obsah CD

Na CD jsou uloženy 2 složky tests a doc.

- **tests -** obsahuje 5 podsložek s implementovanými testy pro jednotlivé balíčky. Podsložky jsou nazvány podle testovaných balíčků `autofs`, `samba`, `krb5`, `nss-pam-ldapd` a `openldap-clients`. Každá podsložka obsahuje konfigurační soubory a certifikáty potřebné o test. Součástí každé podsložky je také testovací skript `runtest.sh` pro Bash využívající knihovnu BeakerLib a soubor `PURPOSE` s popisem testu. Pro spuštění testů je nutné nainstalovat balíček BeakerLib, ostatní potřebné balíčky si již testy v případě potřeby nainstalují automaticky. Testy lze spustit příkazem `bash runtest.sh`.
- **doc -** obsahuje text této bakalářské práce a zdrojové soubory pro vytvoření textu práce.